

EXHIBIT 8

Received: from aegean.openmarket.com (aegean.openmarket.com [199.170.183.5]) by relay.openmarket.com (8.6.10/8.6.6) with ESMTP id VAA08449; Wed, 3 May 1995 21:10:22 -0400
Received: from OpenMarket.com (localhost [127.0.0.1]) by aegean.openmarket.com (8.6.10/8.6.6) with ESMTP id VAA02855; Wed, 3 May 1995 21:10:22 -0400
Message-Id: <199505040110.VAA02855@aegean.openmarket.com>
To: "Lawrence C. Stewart" <stewart@OpenMarket.com>
cc: tech@OpenMarket.com, payne@OpenMarket.com
Subject: Re: Netscape Navigator session capability
In-reply-to: Your message of "Wed, 03 May 1995 16:50:17 EDT."
<199505032050.QAA01977@arctic.openmarket.com>
Date: Wed, 03 May 1995 21:10:21 -0400
From: Andrew Payne <payne@OpenMarket.com>

>There is a persistent rumor that there is a way to have a Netscape
>Navigator save away some packet of information and represent it to the
>server later.
>Does anyone know about this?

I tried a little poking around and "found" that if the server sends back

Set-cookie: <string>

the Netscape client will cache this <string> and present it back to the
server in the form:

Cookie: <string>

for subsequent accesses to that URL. It appears to use the same caching rules
that are used for URLs for username/password access.

It doesn't appear to be documented anywhere, but a search through the
www-talk list found a message by "Lou Montulli" <montulli@netscape.com>,
dated April 18, attached, that seems to spill the beans.

-andy

Path: brighton.openmarket.com!nobody
Date: Tue, 18 Apr 1995 18:18:13 +0500
Message-ID: <9504181233.ZM27022@strumpet.mcom.com>
From: "Lou Montulli" <montulli@netscape.com>
To: Multiple recipients of list <www-talk@www10.w3.org>
Subject: Re: Session tracking
Distribution: local
Newsgroups: oml.mail.www-talk
Approved: news@openmarket.com
Lines: 126

On Apr 17, 10:43pm, Brian Behlendorf wrote:

> Subject: Session tracking
>
> So, I'd like to propose for discussion a new HTTP header (hi Roy!) called
> "Session-ID". This would be optional, of course, and it would change any
> time the browser is restarted (or when the user wished). It would
> consist of a string of 32 random base 64 characters (or whatever encoding
> is allowed in headers). It would allow the content provider to see the
> "path" one takes through his system, even when two separate requests are
> interlacing through a proxy server (HotWired would often get 5
> individuals hitting it from antares.prodigy.com at the same time),
> without requiring user authentication or divulging of any personal
> information. The "From:" header would also work, but it would give away
> information that most would probably prefer not to give.
>

We have been doing some work along these lines. Here
is a proposal for a session based id that is capable
of storing id's accross user sessions. Hopefully
this can be polished up and added to the next HTTP
spec.

Syntax of the Set-Cookie HTTP Response Header:

Set-Cookie: NAME=OPAQUE_STRING \\
[; expires=] \\
[; path=] \\
[; domain=] \\
[; secure]

Syntax of the Cookie HTTP Request Header:

Cookie: NAME=OPAQUE_STRING *[, NAME=OPAQUE_STRING]

Semantics:

The Set-Cookie tag, when sent from server to client as part of any HTTP
response, instructs the client to send the name/value pair using the
"Cookie:" header to the server whenever returning to the specified server
location.

The server location may be specified in several ways:

- o "Set-Cookie" without any optional arguments specifies that all
subsequent requests to the same server within the subtree defined by
the referring URL will include the NAME=OPAQUE_STRING pair. The root
of the subtree is determined by stripping everything after the last
slash. For example, if the referring URL pointed to a file
"/foc/bar/baz.html" the root of the subtree will be defined by

"/foo/bar/".

- o The "path" attribute, if present, specifies which path in the server URL namespace forms the root of the tree for which requests shall include the NAME=OPAQUE_STRING name/value pair. Note that the path is used as a mask value whereby any URL in the same server whose first path characters exactly match the "path" string will match. Therefore the path "/foo" would match "/foobar" and the "/foo/bar.html". The path "/" is the most general path.
- o The "domain" attribute, if present, specifies a server domain in the form of a TCP/IP domain name. Note that the domain acts as a tail end mask. All hosts within the specified domain will receive the cookie on subsequent requests. Only hosts within the specified domain can set a cookie for a domain and domains must have at least two (2) periods in them to prevent domains of the form: ".com" and ".edu". ".mcom.com" is an example of a valid domain.
- o The "expires" attribute specifies an RFC 850 date string that indicates when the cookie should be purged.

Also note that:

- o Multiple "Set-Cookie" headers can be issued in a single server response.
- o Instances of the same path and name will overwrite each other, with the latest instance taking precedence. Instances of the same path but different names will add additional mappings.
- o Setting the path to a higher-level value does not override other more specific path mappings. Therefore you may receive a duplicate name string if other mappings have been made for more specific directories. (See examples below.)
- o The "expires" attribute acts as an indication of when the client can dispose of the "Set-Cookie" mapping. Clients are encouraged to hold the string as long as possible or until the expires date has passed. The expires header lets the client know when it is safe to purge the mapping but the client is not required to do so. Clients should never send a cookie to a server after the expires date has passed. Cookies sent without an expires header are assumed to be short lived and should be purged from the client at the end of the users session. This is typically done when the user quits the client application.
- o The "secure" attribute tells the client to only send the name/value pair when speaking on a secure channel. This protects against the case where the same server has secure and non-secure transactions taking place on the same server and the transmission of the name/value pair could be a security risk. (Note: if the port number of the request is always taken into account when deciding whether to send a name/value pair, this attribute is probably not actually necessary at this time.)
- o The client returns the OPAQUE_STRING to the server in the form of a "Cookie: NAME=OPAQUE_STRING *[:NAME=OPAQUE_STRING]" header. The name/value pairs can be spread among more than one "Cookie" header if the client so chooses.
- o Semi-colons, commas, and whitespace are not allowed in the NAME=OPAQUE_STRING. Servers are expected to use an escaping mechanism to send and read the NAME=OPAQUE_STRING. The '%xx' hex string escaping mechanism used for URLs is recommended (a space character maps to '%20', etc.).
- o When sending cookies to a server, all cookies with a more specific path mapping should be sent before cookies with less specific path mappings. For example, a cookie "name1=foo" with a path mapping of "/" should be sent in front of a cookie "name1=foo2" with a path mapping of "/bar" if they are both to be sent.